AMENDMENTS TO THE CLAIMS

Please amend the claims as follows:

Listing of Claims:

Claim 1 (Currently Amended): A tag privacy protection method for preventing privacy information of a user from being acquired from information which is delivered from a tag device, ~~in which a confidential value corresponding to each tag ID information is stored in a confidential value memory of each tag device;~~ comprising: ~~the steps of the tag device~~

delivering tag output information from an output section of the tag device, the tag output information ~~which~~ correspondings to a confidential value stored in ~~the~~ a confidential value memory of the tag device, the confidential value being a mapping of tag ID information~~from an output section of the each tag device~~;

and

-reading out the confidential value or ~~at least part of~~ elements thereof ~~of the confidential value~~ from the confidential value memory, applying ~~thereto~~ a first function to the confidential value or the elements thereof to generate a new confidential value, an inverse image of the first function~~which~~ being~~is~~ difficult to obtain, and updating the confidential value in the confidential value memory with ~~a result of~~ the new confidential value ~~such calculation~~ by overwriting in a first calculator of the tag device, the new confidential value differing from the confidential value, and both the new confidential value and the confidential value corresponding to the identical tag ID information.

Claim 2 (Currently Amended): A tag privacy protection method according to Claim 1, further comprising:

~~in which a second calculator of the tag device~~ readings out the confidential value from the confidential value memory, and applying~~ies~~ a second function F2 which disturbs a relationship between elements of a definition domain and a mapping thereof to the confidential value read out in a second calculator of the tag device; wherein, ~~and~~ the tag output information is a result of such calculation ~~is the tag output information~~.

Claim 3 (Original): A tag privacy protection method according to Claim 2 in which at least one of the first function F1 and the second function F2 is a hash function.

Claim 4 (Original): A tag privacy protection method according to Claim 2 in which the first function F1 is a hash function H(x)=hash(p | x) where hash represents a hash function for $\{0, 1\}^* \rightarrow \{0, 1\}^r$, $p \in \{0, 1\}^s$ and r and s are natural numbers and in which the second function F2 is a hash function G(x)=hash(q | x) where $q \in \{0, 1\}^s$ for $p \neq q$.

Claim 5 (Original): A tag privacy protection method according to Claim 2 in which the first function F1 is a hash function H(x)=hash(pad(x, p)) where hash represents a hash function for $\{0, 1\}^* \rightarrow \{0, 1\}^r$, $p \in \{0, 1\}^s$, pad(x, p) represents a padding of p to x and r and s are natural numbers and in which the second function F2 is a hash function G(x)=hash(pad(x, q)) where pad(x, q) represents a padding of q to x and $q \in \{0, 1\}^s$ for $p \neq q$.

Claim 6 (Original): A tag privacy protection method according to Claim 2 where the first function F1 is a hash function H(x) for $\{0, 1\}^* \rightarrow \{0, 1\}^r$ and r is a natural number and in which the second function F2 is a hash function G(x)=F(rx) where rx represents a bit inversion of x.

Claim 7 (Original): A tag privacy protection method according to Claim 2 in which at least one of the first function F1 and the second function F2 is a common key encryption function.

Claim 8 (Original): A tag privacy protection method according to Claim 2 in which the first function F1 and the second function F2 are an identical common key encryption function, to which different common keys are applied.

Claim 9 (Currently Amended): A tag privacy protection method for preventing privacy information of a user from being acquired from information which is delivered from a tag device, in which a first confidential value $s_{k,i}$ corresponding to each tag ID information id_k is stored in a confidential value memory of each tag device k where ~~(~~$k \in \{1, \ldots, m\}$~~,~~ and ~~where~~ m represents a total number of tag devices~~,)~~ and in which each tag ID information id_n ($n \in \{1, \ldots, m\}$ and a corresponding second confidential value $s_{n,1}$ are stored in a database memory of a backend apparatus in a manner relating to each other; comprising: ~~the steps of the tag device~~

reading out the first confidential value $s_{k,i}$ from the confidential value memory, and applying a second function F2 which disturbs a relationship between elements of a definition domain and a mapping thereof to generate tag output information $F2(s_{k,i})$ in a second calculator of the tag device;

delivering the tag output information $F2(s_{k,i})$ from an output section of the tag device;

and reading out the first confidential value $s_{k,i}$ from the confidential value memory, applying thereto a first function F1, an inverse image of which is difficult to obtain, and saving a result of such calculation $F1(s_{k,i})$ as new first confidential value $s_{k,i+1}$ in the confidential value memory by overwriting in a first calculator of the tag device;

~~the backend apparatus~~

accepting an input of the tag output information $F2(s_{k,i})$ at an input section of the backend apparatus;

reading out the second confidential value $s_{n,1}$ from the database memory, applying to each second confidential value $s_{n,1}$ read out j times where ~~(~~$j \in \{0, \ldots, j_{max}\}$~~,)~~ the first function F1 and subsequently applying the second function F2 thereto in a third calculator of the backend apparatus;

comparing the tag output information $F2(s_{k,i})$ against the result of calculation $F2(F1^j(s_{n,1}))$ in a comparator of the backend apparatus;

in the event the tag output information $F2(s_{k,i})$ does not match the result of calculation $F2(F1^j(s_{n,1}))$, the processings in the third calculator and the comparator being executed again by changing the value of at least one of n and j;

and extracting by a reader of the backend apparatus the tag ID information id_n which is related to the second confidential value $s_{n,1}$ corresponding to the matched result of calculation $F2(F1^j(s_{n,1}))$ from the database memory when the tag output information $F2(s_{k,i})$ matches the result of calculation $F2(F1^j(s_{n,1}))$.

Claim 10 (Currently Amended): A tag privacy protection method for preventing privacy information of a user from being acquired from information which is delivered from a tag device, in which a first confidential value $s_{k,i}$ and a first proper value w_k corresponding to each tag ID information id_k are sored in a confidential value memory of each tag device k ~~($k \in \{1, \ldots, m\}$,~~ where $k \in \{1, \ldots, m\}$ and m represents a total number of tag devices~~,)~~ in a manner relating to each other and in which each tag ID information id_n ($n \in \{1, \ldots, m\}$) and a corresponding second confidential value $s_{n,1}$ and a second proper value w_n are stored in a

database memory of a backend apparatus in a manner relating to each other; comprising: ~~the steps of~~

~~the tag device~~

reading out the first confidential value $s_{k,i}$ from the confidential value memory and applying thereto a second function F2 which disturbs a relationship between elements of a definition domain and a mapping thereof to generate tag output information $F2(s_{k,i})$ in a second calculator of the tag device;

delivering the tag output information $F2(s_{k,i})$ from an output section of the tag device;

reading out the first confidential value $s_{k,i}$ and the first proper value w_k from the confidential value memory, applying a first function F1, an inverse image of which is difficult to obtain, to a bit combination value of the first confidential value and the first proper value, and saving a result of such calculation $F1(s_{k,i} \mid w_k)$ as a new confidential value $s_{k,i+1}$ in the confidential value memory by overwriting in a first calculator of the tag device;

~~the backend apparatus~~

accepting an input of the tag output information $F2(s_{k,i})$ by an input section of the backend apparatus;

reading out the second confidential value $s_{n,1}$ and the second proper value w_n from the database memory of the backend apparatus, and applying the second function F2 to $I^j(n)$ where $I^j(n)=s_{n,1}$ $(j=0)$ and $I^j(n)=F1(I^{j-1}(n)) \mid id_n)$ $(j \geq 1)$ to calculate $F2(I^j(n))$ in a third calculator of the backend apparatus;

comparing the tag ID information $F2(s_{k,i})$ and a result of calculation $F2(I^j(n))$ in the third calculator in a comparator of the backend apparatus;

in the event the tag output information $F2(s_{k,i})$ does not match the result of calculation $F2(I^j(n))$, the processings in the third calculator and the comparator being executed again by changing the value of at least one of n and j;

and in the event the tag output information $F2(s_{k,i})$ matches the result of calculation $F2(I^j(n))$, extracting the tag ID information id_n which is related to the second confidential value $s_{n,l}$ and the second proper value w_n corresponding to the matching result of calculation $F2(I^j(n))$ from the database memory by a reader.

Claim 11 (Currently Amended): A tag privacy protection method for preventing privacy information of a user from being acquiring from information which is delivered from a tag device, in which a first confidential value $s_{k,i}$ and a first proper value w_k which correspond to each tag ID information id_k are stored in a confidential value memory of each tag device k ~~($k \in \{1, \ldots, m\},$~~ where $k \in \{1, \ldots, m\}$ and m represents a total number of tag devices~~,)~~ and in which a tag ID information id_n ($n \in \{1, \ldots, m\}$) and a second confidential value $s_{n,l}$ and a second proper value w_n which correspond thereto are stored in a database memory of a backend apparatus in a manner relating to each other; comprising: ~~the steps of the tag device~~

reading out the first confidential value $s_{k,i}$ and the first proper value w_k from the confidential value memory and applying to a bit combination value thereof a second function F2 which disturbs a relationship between elements of a definition domain and a mapping thereof to generate tag output information $F2(s_{k,i} \mid w_k)$ in a second calculator of the tag device;

delivering the tag output information $F2(s_{k,i} \mid w_k)$ from an output section of the tag device;

~~and~~ reading out the first confidential value $s_{k,i}$ from the confidential value memory, applying a first function F1, an inverse image of which is difficult to obtain, to the first confidential value $s_{k,i}$ which is read out, and saving a result of such calculation $F1(s_{k,i})$ as a

new first confidential value $s_{k,i}$ in the confidential value memory by overwriting in a first calculator of the tag device;

~~the backend apparatus~~

accepting the tag output information $F2(s_{k,i} \mid w_k)$ as an input at an input section of the backend apparatus ~~at an input section~~;

reading out the second confidential value $s_{n,1}$ and the second proper value w_n from the database memory of the backend apparatus, applying j times ($j \in \{0, \ldots, j_{max}\}$) the first function F1 to the second confidential value $s_{n,1}$ to determine a bit combination value $F1^j(s_{n,i}) \mid w_n$ of a resulting $F1^j(s_{n,i})$ and the second proper value w_n, and applying the second function F2 to the bit combination value $F1^j(s_{n,i}) \mid w_n$ in a third calculator of the backend apparatus; and

comparing the tag output information $F2(s_{k,i} \mid w_k)$ against a result of calculation in the third calculator $F2(F1^j(s_{n,i}) \mid w_n)$ in a comparator of the backend apparatus;

in the event the tag output information $F2(s_{k,i} \mid w_k)$ does not match the result of calculation $F2(F1^j(s_{n,i}) \mid w_n)$, executing the processings in the third calculator and the comparator again by changing the value of at least one of n and j;

and in the event the tag output information $F2(s_{k,i} \mid w_k)$ matches the result of calculation $F2(F1^j(s_{n,i}) \mid w_n)$, extracting the tag ID information id_n which is related to the second confidential value $s_{n,1}$ and the second proper value w_n corresponding to the matching result of calculation $F2(F1^j(s_{n,i}) \mid w_n)$ from the database memory by a reader of the backend apparatus.

Claim 12 (Currently Amended): A tag privacy protection method for preventing privacy information of a user from being acquired from information which is delivered from a tag device, in which a first proper value w_k corresponding to each tag ID information id_k

and a first confidential value s_i which assumes an identical initial value s_1 for a plurality of tag ID information are stored in a confidential value memory of each tag device k where ~~(~~k $\in$ {1, ..., m}~~, where~~ and m represents a total number of tag devices~~)~~, each tag ID information id_n (n $\in$ {1, ..., m}) and a corresponding second proper value w_n are stored in a database memory of a backend apparatus in a manner relating to each other, and a first result of calculation s_{j+1} obtained by applying j times (j $\in$ {0, ..., j_{max}}) a first function F1 to the second confidential value s_1 which is used in common by the plurality of tag ID information is stored in a calculated value memory of the backend apparatus; comrprising: ~~the steps of~~

~~the tag device~~

reading out the first confidential value s_i and the first proper value w_k from the confidential value memory and applying to a bit combination value thereof a second function F2 which disturbs a relationship between elements of a definition domain and a mapping thereof to generate tag output information $F2(s_i \mid w_k)$ in a second calculator of the tag device;

delivering the tag output information $F2(s_i \mid w_k)$ from an output section of the tag device;

~~and~~ reading out the first confidential value s_i from the confidential value memory, applying the first function F1, an inverse image of which is difficult to obtain, to the first confidential value s_i which is read out, and saves a result of such calculation $F1(s_i)$ as a new first confidential value s_{i+1} in the confidential value memory by overwriting in a first calculator of the tag device;

~~the backend apparatus~~

accepting the tag output information $F2(s_i \mid w_k)$ as an input at an input section of the backend apparatus;

reading out a result of the first calculation s_{j+1} and the second proper value w_n from the database memory to obtain a bit combination value $s_{j+1} \mid w_n$ thereof, and applying the second function F2 thereto in a third calculator of the backend apparatus; and

comparing the tag output information $F2(s_i \mid w_k)$ against a result of the calculation by the third calculator $F2(s_{j+1} \mid w_n)$ in a comparator of the backend apparatus;

in the event the tag output information $F2(s_i \mid w_k)$ does not match the result of the calculation $F2(s_{j+1} \mid w_n)$, executing the processings in the third calculator and the comparator again by changing the value of at least one of n and j;

and in the event the tag output information $F2(s_i \mid w_k)$ matches the result of the calculation $F2(s_{j+1} \mid w_n)$, extracting the tag ID information id_n which is related to the second proper value w_n corresponding to the matching result of calculation $F2(s_{j+1} \mid w_n)$ from the database memory by a reader of the backend apparatus.

Claim 13 (Currently Amended): A tag privacy protection method for preventing privacy information of a user from being acquired from information which is delivered from a tag device, in which a combination of d ($d \geq 2$) elements $e_{u, vu}$ ($u \in \{1, \ldots, d\}$) corresponding to each tag ID information id_k is stored in a confidential value memory of each tag device k where ~~(~~$k \in \{1, \ldots, m\}$~~;~~ and ~~where~~ m represents a total number of tag devices)~~,~~ and in which a combination of d initial elements $f_{u, 0}$ comprising one selected from each of d kinds ($d \geq 2$) of subgroups α_u ($u \in \{1, \ldots, d\}$) and the tag ID information id_n of each tag device n ($n \in \{1, \ldots, m\}$) are stored in a database memory of a backend apparatus in a manner relating to each other~~,~~ comprising: ~~the steps of;~~

~~the tag device~~

reading out the d elements $e_{u, vu}$ from the confidential value memory to form a bit combination value thereof which represents a confidential value $s_{k, i}$ and applying a second

function F2 which disturbs a relationship between elements of a definition domain and a mapping thereof to the confidential value $s_{k,i}$ to generate tag output information $a_{k,i}=F2(s_{k,i})$ in a second calculator of the tag device;

delivering the tag output information $a_{k,i}$ from an output section of the tag device;

~~and~~ extracting at least part of elements $e_{u',vu'}$ ($u' \in \{1, \ldots, d\}$) from the confidential value memory, applying a first function F1, an inverse image of which is difficult to obtain, to the extracted elements $e_{u',vu'}$, and saving a result of such calculation $F1(e_{u',vu'})$ as new elements $e_{u',vu'+1}$ in the confidential value memory by overwriting in a first calculator of the tag device;

~~the backend apparatus~~

accepting the tag output information $a_{k,i}$ as an input at an input section of the backend apparatus;

applying the first function F1 w_u times ($w_u \in \{1, 2, \ldots, max\}$) to d initial elements $f_{u,0}$ ($u \in \{1, \ldots, d\}$) corresponding to the tag ID information id_n, and applying the second function F2 to a bit combination value of these values $F1^{wu}(f_{u,0})$ to determine a calculated value c in a third calculator of the backend apparatus; and

comparing the tag output information $a_{k,i}$ against the calculated value c in a comparator of the backend apparatus;

in the event the tag output information $a_{k,i}$ does not match the calculated value c, executing the processings in the third calculator and the comparator again by changing the value of at least part of n and w_u ;

and in the event the tag output information $a_{k,i}$ matches the calculated value c, extracting tag ID information id_n which is related to the combination of d initial elements $f_{u,0}$ corresponding to the calculated value c from the database memory by a reader of the backend apparatus.

Claim 14 (Currently Amended): A tag privacy protection method for preventing privacy information of a user from being acquired from information which is delivered from a tag device, in which a combination of d ($d \geq 2$) elements $e_{u, vu}$ ($u \in \{1, \ldots, d\}$) which corresponds to each tag ID information id_k and a proper value γ_k which is inherent to each tag ID information id_k are stored in a confidential value memory of each tag device k where ~~($k \in \{1, \ldots, m\}$ and, where~~ m represents a total number of tag devices~~,)~~ and in which a combination of d ($d \geq 2$) elements $e_{u, vu}$ where ~~($u \in \{1, \ldots, d\}$,)~~ which corresponds to each tag ID information id_k and a proper value γ_k which is inherent to each tag ID information id_k are stored in a database memory of a backend apparatus in a manner relating to each other; comprising: ~~the steps of~~

~~the tag device~~

reading out the d elements $e_{u, vu}$ and the proper value γ_k from the confidential value memory, and applying a second function F2 which disturbs a relationship between elements of a definition domain and a mapping thereof to a confidential value $s_{k, i}$ which is a bit combination value of the d elements and the proper value to generate tag output information $a_{k, i}=F2(s_{k, i})$ in a second calculator of the tag device;

delivering the tag output information $a_{k, I}$ from an output section of the tag device;

~~and~~ extracting at least part of elements $e_{u', vu'}$ ($u' \in \{1, \ldots, d\}$) from the confidential value memory, applying a first function F1, an inverse image of which is difficult to obtain, to the extracted elements $e_{u', vu'}$, and saving a result of such calculation F1 ($e_{u', vu'}$) as new elements $e_{u', vu'+1}$ in the confidential value memory by overwriting in a first calculator of the tag device;

~~the backend apparatus~~

accepting the tag output information $a_{k,i}$ as an input at an input section of the backend apparatus;

applying the first function F1 w_u times ($w_u \in \{1, 2, \ldots, \max\}$) to the d initial elements $f_{u,0}$ ($u \in \{1, \ldots, d\}$) corresponding to the tag ID information id_n and applying the second function F2 to a bit combination value of the function values $F1^{wu}(f_{u,0})$ and the proper value γ_n to determine a calculated value c in a third calculator of the backend apparatus; and

comparing the tag output information $a_{k,i}$ against the calculated value c in a comparator of the backend apparatus;

in the event the tag output information $a_{k,i}$ does not match the calculated value c, executing the processings in the third calculator and the comparator again by changing the value of al least part of n and w_u;

and in the event the tag output information $a_{k,i}$ matches the calculated value c, extracting tag ID information id_n which is related to the combination the plurality of initial elements $f_{u,0}$ corresponding to the calculated value c from the database memory by a reader of the backend apparatus.

Claim 15 (Currently Amended): A tag privacy protection method for preventing privacy information of a user from being acquired from information which is delivered from a tag device, in which d ($d \geq 1$) elements $e_{u,vu}$ ($u \in \{1, \ldots, d\}$) are stored in a confidential value memory of each tag device k where ~~(~~$k \in \{1, \ldots, m\}$~~,~~ and ~~where~~ m represents a total number of tag devices~~)~~, a manifold value z having t kinds ($t \geq 2$) of values is stored in a first manifold value memory of each tag device k, a combination of d initial elements $f_{u,0}$ comprising one selected from each of d kinds ($d \geq 1$) of subgroups α_u ($u \in \{1, \ldots, d\}$) and tag ID information id_n ($n \in \{1, \ldots, m\}$) of each tag device are stored in a database memory of a

backend apparatus in a manner relating to each other, and the manifold value z is stored in a second manifold value memory of the backend apparatus; comprising: ~~the steps of~~

~~the tag device~~

reading out each element $e_{u, vu}$ from the confidential value memory and reading out either manifold value z from the first manifold value memory and applying a second function F2 which disturbs a relationship between elements of a definition domain and a mapping thereof to a confidential value $s_{k, i}$ which is a bit combination value of the elements and the manifold value to generate tag output information $a_{k, i}=F2(s_{k, i})$ in a second calculator of the tag device;

delivering the tag output information $a_{k, i}$ from an output section of the tag device;

~~and~~ extracting at least part of elements $e_{u', vu'}$ ($u' \in \{1, \ldots, d\}$) from the confidential value memory each time the output section delivers the tag output information $a_{k, i}$ t times, applying a first function F1, an inverse image of which is difficult to obtain, to the extracted elements $e_{u', vu'}$, and saving a result of such calculation $F1(e_{u', vu'})$ as new elements $e_{u', vu'+1}$ in the confidential value memory by overwriting in a first calculator of the tag device;

~~the backend apparatus~~

accepting the tag output information $a_{k, i}$ as an input at an input section of the backend apparatus;

applying the first function F1 w_u times ($w_u \in \{1, 2, \ldots, max\}$) to the d initial elements $f_{u, 0}$ ($u \in \{1, \ldots, d\}$) corresponding to the tag ID information id_n and applying the second function F2 to a bit combination value of these values $F1^{wu}(f_{u, 0})$ and the manifold value z to determine a calculated value c in a third calculator of the backend apparatus; and

comparing the tag output information $a_{k, i}$ against the calculated value c in a comparator of the backend apparatus;

in the event the tag output information $a_{k,i}$ does not match the calculated value c, executing the processings in the third calculator and the comparator again by changing the value of at least part of n, w_u and z;

and in the event the tag output information $a_{k,i}$ matches the calculated value c, extracting the tag ID information id_n which is related to the combination of the d initial elements $f_{u,0}$ corresponding to the calculated value c from the database memory by a reader of the backend apparatus.

Claim 16 (Currently Amended): A tag privacy protection method for preventing privacy information of a user from being acquired from information which is delivered from a tag device, in which d ($d \geq 2$) elements $e_{u,vu}$ ($u \in \{1, \ldots, d\}$) are stored in a confidential value memory of each tag device k ($k \in \{1, \ldots, m\}$, where m represents a total number of tag devices), a manifold value z_u which assumes t_u kinds ($t_u \geq 2$) of values for each u is stored in a first manifold value memory of each tag device k, a combination of d initial elements $f_{u,0}$ comprising one selected from each of d kinds ($d \geq 2$) of subgroups α_u ($u \in \{1, \ldots, d\}$) and tag ID information id_n ($n \in \{1, \ldots, m\}$) of each tag device are stored in a database memory of a backend apparatus in a manner relating to each other, and the manifold value z_u is stored in a second manifold value memory of the backend apparatus; comprising: ~~the steps of~~

~~the tag device~~

reading out each element $e_{u,vu}$ from the confidential value memory and reading out either manifold value z_u for each u from the first manifold value memory and applying a second function F2 which disturbs a relationship between elements of a definition domain and a mapping thereof to a confidential value $s_{k,i}$ which is a bit combination value of $e_{u,vu}$ and z_u to generate tag output information $a_{k,i}=F2(s_{k,i})$ in a second calculator of the tag device;

delivering the tag output information $a_{k,i}$ from an output section of the tag device;

extracting at least part of elements $e_{u', vu'}$ ($u' \in \{1, \ldots, d\}$) from the confidential value memory each time the output section delivers the tag output information $a_{k, i}$ some number of times, applying a first function F1, an inverse image of which is difficult to obtain, to the extracted elements $e_{u', vu'}$, and saving a result of such calculation $F1(e_{u', vu'})$ as new elements $e_{u', vu'+1}$ in the confidential value memory by overwriting in a first calculator of the tag device;

~~the backend apparatus~~

accepting the tag output information $a_{k, i}$ as an input at an input section of the backend apparatus;

applying w_u times ($w_u \in \{1, 2, \ldots, max\}$) the first function F1 to the d initial elements $f_{u, 0}$ ($u \in \{1, \ldots, d\}$) corresponding to the tag ID information id_n, and applying the second function F2 to a bit combination value of these values $F1^{wu}(f_{u, 0})$ and the manifold value z_u to determine a calculated value c in a third calculator of the backend apparatus; and

comparing the tag output information $a_{k, i}$ against the calculated value c in a comparator of the backend apparatus;

in the event the tag output information $a_{k, i}$ does not match the calculated value c, executing the processings in the third calculator and the comparator again by changing the value of at least part of n, w_u and z_u;

and in the event the tag output information $a_{k, i}$ matches the calculated value c, extracting tag ID information id_n which is related to the combination of a plurality of initial elements $f_{u, 0}$ corresponding to the calculated value c from the database memory by a reader of the backend apparatus.

Claim 17 (Currently Amended): A tag device for use in an automatic tag identification system comprising:

a confidential value memory in which a confidential value corresponding to tag ID information is stored;

a second calculator connected to the confidential value memory for reading out the confidential value from the confidential value memory and for applying a second function F2 which disturbs a relationship between elements of a definition domain and a mapping thereof to the confidential value which is read out to generate tag output information;

an output section for delivering the tag output information;

and a first calculator for reading out at least part of elements of the confidential value from the confidential value memory and for applying a first function F1, an inverse image~~a mapping~~ of which is difficult to obtain, to the elements which are read out, with a result of such calculation being used to update the confidential value in the confidential value memory by overwriting.

Claim 18 (Currently Amended): A backend apparatus for use in an automatic tag identification system comprising

a database memory in which each tag ID information and a corresponding confidential value are related to each other;

an input section which accepts tag output information as an input;

a calculator for applying a first function F1 which is used in a tag device some number of times to at least part of elements of the confidential value in the database memory and which then applies a second function which is used in the tag device thererto;

a comparator for sequentially comparing a result of the calculation in the calculator against the tag output information;

and a reader for extracting the tag ID information which is related to the confidential value corresponding to the matching result of calculation when a matching between the result of calculation and the tag output information is found from the database memory, wherein

an inverse image of the first function F1 is difficult to obtain, the second function F2 disturbs a relationship between elements of a definition domain and a mapping thereof.

Claim 19 (Original): A tag device for use in an automatic tag identification system comprising

a confidential value memory in which a first confidential value $s_{k,i}$ corresponding to tag ID information id_k is stored;

a second calculator connected to the confidential value memory for reading out the first confidential value $s_{k,i}$ from the confidential value memory and for applying a second function F2 which disturbs a relationship between elements of a definition domain and a mapping thereof to the first confidential value $s_{k,i}$ to generate tag output information $F2(s_{k,i})$;

an output section for delivering the tag output information $F2(s_{k,i})$;

and a first calculator connected to the confidential value memory for reading out the first confidential value $s_{k,i}$ from the confidential value memory, for applying a first function F1, an inverse image of which is difficult to obtain, to the first confidential value and for saving a result of such calculation $F1(s_{k,i})$ as a new first confidential value $s_{k,i+1}$ in the confidential value memory by overwriting.

Claim 20 (Original): A tag device according to Claim 19, further comprising

a counter for counting a number of times rn the first confidential value is updated,

the output section also delivering information which specifies the number of updating times rn.

Claim 21 (Original): A tag device for use in an automatic tag identification system comprising

a confidential value memory in which a first confidential value $s_{k,i}$ and a first proper value w_k which correspond to a tag ID information id_k are stored;

a second calculator connected to the confidential value memory for reading out the first confidential value $s_{k,i}$ from the confidential value memory and for applying a second function F2 which disturbs a relationship between elements of a definition domain and a mapping thereof to the first confidential value to generate tag output information $F2(s_{k,i})$;

an output section for delivering the tag output information $F2(s_{k,i})$;

and a first calculator connected to the confidential value memory for reading out the first confidential value $s_{k,i}$ and the first proper value w_k from the confidential value memory, for applying a first function F1, an inverse image of which is difficult to obtain, to a bit combination value of the first confidential value and the first proper value and for saving a result of such calculation $F1(s_{k,i} \mid w_k)$ as a new first confidential value $s_{k,i+1}$ in the confidential value memory by overwriting.

Claim 22 (Currently Amended): A backend apparatus for use in an automatic tag identification system comprising

a database memory in which each tag ID information id_n ($n \in \{1, \ldots, m\}$, where m represents a total number of tag devices) and a second confidential value $s_{n,1}$ corresponding thereto are related to each other;

an input section which accepts tag output information $F2(s_{k,i})$ as an input;

a third calculator connected to the database memory for reading out the second confidential value $s_{n,1}$ from the database memory, applying j times ($j \in \{0, \ldots, j_{max}\}$) a first

function F1 which is used in a tag device to each of the second confidential values $s_{n,1}$ which are read out, and for subsequently applying a second function F2 which is used in the tag device;

a comparator for comparing the tag output information $F2(s_{k,i})$ against a result of calculation in the third calculator $F2(F1^j(s_{n,1}))$;

a controller for causing the processings in the third calculator and the comparator to be executed again by changing the value of at least one of n and j in the event the tag output information $F2(s_{k,i})$ and the result of calculation $F2(F1^j(s_{n,1}))$ do not match;

and a reader connected to the database memory and operative when the tag output information $F2(s_{k,i})$ matches the result of the calculation $F2(F1^j(s_{n,1}))$ to extract the tag ID information id_n which is related to the second confidential value $s_{n,1}$ corresponding to the matching result of the calculation $F2(F1^j(s_{n,1}))$ from the database memory, wherein

an inverse image of the first function F1 is difficult to obtain, the second function F2 disturbs a relationship between elements of a definition domain and a mapping thereof.

Claim 23 (Original): A backend apparatus according to Claim 22 in which the input section accepts an input of information which specifies a number of times rn the first confidential value is updated in the tag device, the third calculator applies the first function F1 j=rn times to each of the confidential values $s_{n,1}$ which are read out and then applies the second function F2 thereto, and the controller causes the processings in the third calculator and the comparator to be executed again by changing the value of n when the tag output information $F2(s_{k,i})$ does not match the result of the calculation $F2(F1^j(s_{n,1}))$.

24 (Original): A backend apparatus according to Claim 22 in which the database memory stores the result of the calculation $F2(F1^j(s_{n,1}))$ in the third calculator in a manner

relating it to the second confidential value $s_{n,1}$, and the comparator performs a comparing processing by using the result of the calculation $F2(F1^j(s_{n,1}))$ stored in the database memory.

Claim 25 (Currently Amended): A backend apparatus for use in an automatic tag identification system comprising

a database memory in which each tag ID information id_n ($n \in \{1, \ldots, m\}$), a corresponding second confidential value $s_{n,1}$ and second proper value w_n are stored in a manner relating to each other;

a input section which accepts an input of tag output information $F2(s_{k,i})$;

a third calculator connected to the database memory for reading out the second confidential value $s_{n,1}$ and the second proper value w_n from the database memory and for applying a second function F2 to $I^j(n)$ where $I^j(n)=s_{n,1}$ ($j=0$), and $I^j(n)=F1(I^{j-1}(n) \mid id_n)$ ($j \geq 1$) to calculate $F2(I^j(n))$ where F1 represents a first function;

a comparator for comparing the tag output information $F2(s_{k,i})$ against the result of the calculation in the third calculator $F2(I^j(n))$;

a controller for causing the processings in the third calculator and the comparator to be executed again by changing the value of at least one of n and j when the tag output information $F2(s_{k,i})$ does not match the result of the calculation $F2(I^j(n))$;

and a reader for extracting tag ID information id_n which is related to the second confidential value $s_{n,1}$ and the second proper value w_n corresponding to the matched result of calculation $F2(I^j(n))$ from the database memory when a matching between the tag output information $F2(s_{k,i})$ and the result of the calculation $F2(I^j(n))$ is found wherein

an inverse image of the first function F1 is difficult to obtain, the second function F2 disturbs a relationship between elements of a definition domain and a mapping thereof.

Claim 26 (Original): A tag device for use in an automatic tag identification system comprising

a confidential value memory in which a first confidential value $s_{k,i}$ and a first proper value w_k corresponding to tag ID information id_k are stored;

a second calculator connected to the confidential value memory for reading out the first confidential value $s_{k,i}$ and the first proper value w_k from the confidential value memory and for applying a second function F2 which disturbs a relationship between elements of a definition domain and a mapping thereof to a bit combination value of the first confidential value and the first proper value to generate tag output information $F2(s_{k,i} \mid w_k)$;

an output section for delivering the tag output information $F2(s_{k,i} \mid w_k)$

and a first calculator connected to the confidential value memory for reading the first confidential value $s_{k,i}$ from the confidential value memory, applying a first function F1, an inverse image of which is difficult to obtain, to the first confidential value $s_{k,i}$ which is read out and saving a result of such calculation $F1(s_{k,i})$ as a new first confidential value $s_{k,i+1}$ in the confidential value memory by overwriting.

Claim 27 (Currently Amended): A backend apparatus for use in an automatic tag identification system comprising

a database memory in which each tag ID information id_n ($n \in \{1, \ldots, m\}$) and a corresponding second confidential value $s_{n,1}$ and second proper value w_n are stored in a manner relating to each other;

an input section which accepts an input of tag output information $F2(s_{k,i} \mid w_k)$;

a third calculator connected to the database memory for reading out the second confidential value $s_{n,1}$ and the second proper value w_n from the database memory, applying j times ($j \in \{0, \ldots, j_{max}\}$) a first function F1 which is used in a tag device to the second

confidential value $s_{n,1}$, determining a bit combination value $F1^j(s_{n,i}) \mid w_n$ of a result of application $F1^j(s_{n,i})$ and the second proper value w_n, and applying a second function F2 which is used in the tag device to the bit combination value $F1^j(s_{n,i} \mid w_n)$;

a comparator for comparing the tag output information $F2(s_{k,i} \mid w_k)$ against a result of calculation in the third calculator $F2(F1^j(s_{n,i}) \mid w_n)$;

a controller for causing the processings in the third calculator and the comparator to be executed again by changing the value of at least one of n and j when the tag output information $F2(s_{k,i} \mid w_k)$ does not match the result of the calculation $F2(F1^j(s_{n,i}) \mid w_n)$;

and a reader connected to the database memory for extracting the tag ID information id_n which is related to the second confidential value $s_{n,1}$ and the second proper value w_n corresponding to the matched result of calculation $F2(F1^j(s_{n,i}) \mid w_n)$ when a matching between the tag output information $F2(s_{k,i} \mid w_k)$ and the result of the calculation $F2(F1^j(s_{n,i}) \mid w_n)$ is found wherein

an inverse image of the first function F1 is difficult to obtain, the second function F2 disturbs a relationship between elements of a definition domain and a mapping thereof.

Claim 28 (Original): A tag device for use in an automatic tag identification system comprising

a confidential value memory in which a first proper value w_k corresponding to each tag ID information id_k and a first confidential value s_i which assumes an equal initial value s_1 for a plurality of tag ID information are stored;

a second calculator connected to the confidential value memory for reading out the first confidential value s_i and the first proper value w_k from the confidential value memory and for applying a second function F2 which disturbs a relationship between elements of a

definition domain and a mapping thereof to a bit combination value of the first confidential value and the first proper value to generate tag output information $F2(s_i \mid w_k)$;

an output section for delivering the tag output information $F2(s_i \mid w_k)$;

and a first calculator connected to the confidential value memory for reading out the first confidential value s_i from the confidential value memory, applying a first function F1, an inverse image of which is difficult obtain, to the first confidential value s_i which is read out and saving a result of such calculation $F1(s_i)$ as a new first confidential value s_{i+1} in the confidential value memory by overwriting.

Claim 29 (Currently Amended): A backend apparatus for use in an automatic tag identification system comprising

a database memory in which each tag ID information id_n ($n \in \{1, \ldots, m\}$) and a corresponding second proper value w_n are stored in a manner relating to each other;

a calculated value memory in which first results of calculation s_{j+1} are stored which are obtained by applying j times ($j \in \{0, \ldots, j_{max}\}$) a first function which is used in a tag device to a second confidential value s_1 which is used in common for a plurality of tag ID information;

an input section which accepts an input of tag output information $F2(s_i \mid w_k)$;

a third calculator connected to the database memory for reading out the first result of calculation s_{j+1} and the second proper value w_n from the database memory to obtain a bit combination value thereof $s_{j+1} \mid w_n$ and for applying a second function F2 which is used in the tag device thereto;

a comparator for comparing the tag output information $F2(s_i \mid w_k)$ and the result of calculation in the third calculator $F2(s_{j+1} \mid w_n)$;

a controller for causing the processings in the third calculator and the comparator to be executed again by changing the value of at least one of n and j when the tag output information $F2(s_i \mid w_k)$ does not match the result of calculation $F2(s_{j+1} \mid w_n)$;

and a reader connected to the database memory for extracting the tag ID information id_n which is related to the second proper value w_n corresponding to the matched result of calculation $F2(s_{j+1} \mid w_n)$ when a matching between the tag output information $F2(s_i \mid w_k)$ and the result of calculation $F2(s_{j+1} \mid w_n)$ is found wherein

an inverse image of the first function F1 is difficult to obtain, the second function F2 disturbs a relationship between elements of a definition domain and a mapping thereof.

Claim 30 (Original): A tag device for use in an automatic tag identification system comprising

a confidential value memory in which a combination of d ($d \geq 2$) elements $e_{u, vu}$ ($u \in \{1, \ldots, d\}$) which corresponds to each tag ID information id_k is stored;

a second calculator connected to the confidential value memory for reading out the d elements $e_{u, vu}$ from the confidential value memory and for applying a second function F2 which disturbs a relationship between elements of a definition domain and a mapping thereof to a confidential value $s_{k, i}$ which is a bit combination value of the d elements to generate tag output information $a_{k, i}=F2(s_{k, i})$;

an output section for delivering the tag output information $a_{k, i}$;

and a first calculator connected to the confidential value memory for extracting at least part of elements $e_{u', vu'}$ ($u' \in \{1, \ldots, d\}$) from the confidential value memory, for applying a first function F1, an inverse image of which is difficult to obtain, to the extracted elements $e_{u', vu'}$ and for saving a result of such calculation $F1(e_{u', vu'})$ as new elements $e_{u', vu'+1}$ in the confidential value memory by overwriting.

Claim 31 (Currently Amended): A backend apparatus for use in an automatic tag identification system comprising

a database memory in which a combination of d initial elements $f_{u,0}$ comprising one selected from each of d kinds ($d \geq 2$) of subgroups α_u ($u \in \{1, \ldots, d\}$), and tag ID information id_n of each tag device n ($n \in \{1, \ldots, m\}$, where m represents a total number of tag devices) are stored in a manner relating to each other;

an input section for accepting an input of tag output information $a_{k,i}$;

a third calculator for applying w_u times ($w_u \in \{1, 2, \ldots, max\}$) a first function F1 to the d initial elements $f_{u,0}$ ($u \in \{1, \ldots, d\}$) which correspond to the tag ID information id_n and for applying a second function F2 to a bit combination value of these values $F1^{wu}(f_{u,0})$ to determine a calculated value c;

a comparator for comparing the tag output information $a_{k,i}$ against the calculated value c;

a controller for causing the processings in the third calculator and the comparator to be executed again by changing the value of at least part of n and w_u when the tag output information $a_{k,i}$ does not match the calculated value c;

and a reader connected to the database memory for extracting tag ID information id_n which is related to the combination of d initial elements $f_{u,0}$ corresponding to the calculated value c when the tag output information $a_{k,i}$ matches the calculated value c, wherein

an inverse image of the first function F1 is difficult to obtain, the second function F2 disturbs a relationship between elements of a definition domain and a mapping thereof.

Claim 32 (Original): A tag device for use in an automatic tag identification system comprising

a confidential value memory in which a combination of d ($d \geq 2$) elements $e_{u, vu}$ ($u \in \{1, \ldots, d\}$) which correspond to each tag ID information id_k and a proper value γ_k which is inherent to each tag ID information id_k are stored;

a second calculator connected to the confidential value memory for reading out the d elements $e_{u, vu}$ and the proper value γ_k from the confidential value memory and for applying a second function F2 which disturbs a relationship between elements of a definition domain and a mapping thereof to a confidential value $s_{k, i}$ which is a bit combination value of the d elements and the proper value to generate tag output information $a_{k, i}=F2(s_{k, i})$;

an output section for delivering the tag output information $a_{k, i}$;

and a first calculator connected to the confidential value memory for extracting at least part of the elements $e_{u', vu'}$ ($u' \in \{1, \ldots, d\}$) from the confidential value memory, applying a first function F1, an inverse image of which is difficult to obtain, to the extracted elements $e_{u', vu'}$ and for saving a result of such calculation $F1(e_{u', vu'})$ as new elements $e_{u', vu'+1}$ in the confidential value memory by overwriting;

Claim 33 (Currently Amended): A backend apparatus for use in an automatic tag identification system comprising

a database memory in which a combination of d initial elements $f_{u, 0}$ comprising one selected from each of d kinds ($d \geq 2$) of subgroups α_u ($u \in \{1, \ldots, d\}$), a proper value γ_n which is inherent to each tag ID information id_n ($n \in \{1, \ldots, m\}$) and each tag ID information id_n are stored in a manner relating to each other;

an input section for accepting an input of tag output information $a_{k, i}$;

a third calculator for applying w_u times ($w_u \in \{1, 2, \ldots, max\}$) a first function F1 to the d initial elements $f_{u, 0}$ ($u \in \{1, \ldots, d\}$) corresponding to the tag ID information id_n and for

applying a second function F2 to a bit combination value of these values $F1^{wu}(f_{u,0})$ and the proper value γ_n to determine a calculated value c;

a comparator for comparing the tag output information $a_{k,i}$ against the calculated value c;

a controller for causing the processings in the third calculator and the comparator to be executed again by changing the value of at least part of n and w_u when the tag output information $a_{k,i}$ does not match the calculated value c;

and a reader connected to the database memory for extracting tag ID information id_n which is related to the combination of a plurality of initial elements $f_{u,0}$ corresponding to the calculated value c from the database memory when a matching between the tag output information $a_{k,i}$ and the calculated value c is found, wherein

an inverse image of the first function F1 is difficult to obtain, the second function F2 disturbs a relationship between elements of a definition domain and a mapping thereof.

Claim 34 (Original): A tag device for use in an automatic tag identification system comprising

a confidential value memory in which d ($d \geq 1$) elements $e_{u,vu}$ ($u \in \{1, \ldots, d\}$) are stored;

a first manifold value memory in which a manifold value z which assumes t kinds ($t \geq 2$) of values is stored;

a second calculator connected to the confidential value memory and the first manifold value memory for reading out the elements $e_{u,vu}$ from the confidential value memory and for reading out either manifold value z from the first manifold value memory and for applying a second function F2 which disturbs a relationship between elements of a definition domain

and a mapping thereof to a confidential value $s_{k,i}$ which is a bit combination value of the elements and the manifold value to generate tag output information $a_{k,i}=F2(s_{k,i})$;

an output section for delivering the tag output information $a_{k,i}$;

and a first calculator connected to the confidential value memory for extracting at least part of elements $e_{u',vu'}$ ($u' \in \{1, \ldots, d\}$) from the confidential value memory each time the output section delivers the tag output information $a_{k,i}$ t times, for applying a first function F1, an inverse image of which is difficult to obtain, to the extracted elements $e_{u',vu'}$ and for saving a result of such calculation $F1(e_{u',vu'})$ as new elements $e_{u',vu'+1}$ in the confidential value memory by overwriting .

Claim 35 (Original): A tag device according to Claim 34 in which as long as the first calculator does not update elements in the confidential value memory, the manifold value z used by the second calculator in generating the tag output information $a_{k,i}$ changes each time the tag output information $a_{k,i}$ is generated.

Claim 36 (Currently Amended): A backend apparatus for use in an automatic tag identification system comprising

a database memory in which a combination of d initial elements $f_{u,0}$ comprising one selected from each of d kinds ($d \geq 1$) of subgroup α_u ($u \in \{1, \ldots, d\}$) and a tag ID information id_n ($n \in \{1, \ldots, m\}$) of each tag device are stored in a manner relating to each other;

a second manifold value memory in which a manifold value z which assumes t kinds ($t \geq 2$) of values is stored;

an input section for accepting an input of tag output information $a_{k,i}$;

a third calculator for applying w_u times ($w_u \in \{1, 2, \ldots, max\}$) a first function F1 to the d initial elements $f_{u,0}$ ($u \in \{1, \ldots, d\}$) in the database memory which correspond to the tag

ID information id_n and for applying a second function F2 to a bit combination value of these values $F1^{wu}(f_{u,0})$ and the manifold value z in the second manifold value memory to determine a calculated value c;

a comparator for comparing the tag output information $a_{k,i}$ against the calculated value c;

a controller for causing the processings in the third calculator and the comparator to be executed again by changing the value at least part of n, w_u and z when the tag output information $a_{k,i}$ does not match the calculated value c;

and a reader connected to the database memory for extracting the tag ID information id_n which is related to the combination of d initial elements $f_{u,0}$ corresponding to the calculated value c from the database memory when a matching between the tag output information $a_{k,i}$ and the calculated value c is found, wherein

an inverse image of the first function F1 is difficult to obtain, the second function F2 disturbs a relationship between elements of a definition domain and a mapping thereof.

Claim 37 (Original): A tag device for use in an automatic tag identification system comprising

a confidential value memory in which d ($d \geq 2$) elements $e_{u,vu}$ ($u \in \{1, \ldots, d\}$) are stored;

a first manifold value memory in which a manifold value z_u which assumes t_u kinds ($t_u \geq 2$) of values for each u is stored;

a second calculator connected to the confidential value memory and the first manifold value memory for reading out the elements $e_{u,vu}$ from the confidential value memory and for reading out either manifold value z_u for each u from the first manifold value memory and for applying a second function F2 which disturbs a relationship between elements of a definition

domain and a mapping thereof to a confidential value $s_{k,i}$ which is a bit combination value of these $e_{v,vu}$ and z_u to generate tag output information $a_{k,i}=F2(s_{k,i})$;

an output section for delivering the tag output information $a_{k,i}$;

and a first calculator connected to the confidential value memory for extracting at least part of the elements $e_{u',vu'}$ ($u' \in \{1, \ldots, d\}$) from the confidential value memory each time the output section delivers the tag output information $a_{k,i}$ some number of times, for applying a first function F1, an inverse image of which is difficult to obtain, to the extracted elements $e_{u',vu'}$, and for saving a result of such calculation $F1(e_{u',vu'})$ as new elements $e_{u',vu'+1}$ in the confidential value memory by overwriting.

Claim 38 (Original): A tag device according to Claim 37 in which each time the output section delivers the tag output information $a_{k,i}$, the first calculator extracts at least part of the elements $e_{u',vu'}$ from the confidential value memory, applies the first function F1 to the extracted elements $e_{u',vu'}$ and saves a result of such calculation $F1(e_{u',vu'})$ as new elements $e_{u',vu'+1}$ in the confidential value memory by overwriting.

Claim 39 (Original): A tag device according to Claim 37 in which each time the output section delivers the tag output information $a_{k,i}$ $\sum_{u=1}^{d} t_u$ times, the first calculator extracts at least part of the elements $e_{u',vu'}$ from the confidential value memory, applies the first function F1 to the extracted elements $e_{u',vu'}$, and saves a result of such calculation $F1(e_{u',vu'})$ as new elements $e_{u',vu'+1}$ in the confidential value memory by overwriting.

40 (Original): A tag device according Claim 39 in which as long as the first calculator does not update the elements in the confidential value memory, a combination of

manifold values z_u ($u \in \{1, \ldots, d\}$) which are used by the second calculator in generating the tag output information $a_{k,i}$ changes each time the tag output information $a_{k,i}$ is generated.

Claim 41 (Currently Amended): A backend apparatus for use in an automatic tag identification system comprising

a database memory in which a combination of d initial elements $f_{u,0}$ which comprises one selected from each of d kinds ($d \geq 1$) of subgroups α_u ($u \in \{1, \ldots, d\}$) and tag ID information id_n ($n \in \{1, \ldots, m\}$) of each tag device are stored in a manner relating to each other;

a second manifold value memory in which a manifold value z_u which assumes t_u kinds ($t_u \geq 2$) of values for each u is stored;

an input section for accepting an input of tag output information $a_{k,i}$;

a third calculator for applying w_u times ($w_u \in \{1, 2, \ldots, max\}$) a first function F1 which is used in a tag device to the d initial elements $f_{u,0}$ ($u \in \{1, \ldots, d\}$) corresponding to the tag ID information id_n and for applying a second function F2 which is used in the tag device to a bit combination value of these values $F1^{wu}(f_{u,0})$ and the manifold value z_u to determine a calculated value c;

a comparator for comparing the tag output information $a_{k,i}$ against the calculated value c;

a controller for causing the processings in the third calculator and the comparator to be executed again by changing the value of at least part of n, w_u and z;

and a reader connected to the database memory for extracting tag ID information id_n which is related to the combination of the d initial elements $f_{u,0}$ corresponding to the calculated value c from the database memory when a matching between the tag output information $a_{k,i}$ and the calculated value c is found, wherein

an inverse image of the first function F1 is difficult to obtain, the second function F2 disturbs a relationship between elements of a definition domain and a mapping thereof.

Claim 42 (Currently Amended): A tag privacy protection method for preventing privacy information of a user from being acquired from information which is delivered from a tag device, in which privileged ID information sid_h which is formed by privileging respective tag ID information id_h is stored in a confidential value memory of each tag device; comprising: ~~the steps of~~

~~the tag device~~

reading out the privileged ID information sid_h stored in the confidential value memory in a read/write section of the tag device;

~~and~~ delivering the privileged ID information sid_h to an updater which is provided externally of each tag device from a first output section of the tag device;

~~the updater~~

accepting an input of the privileged ID information sid_h at a first input section of the updater;

updating the privileged ID information sid_h to ~~generating~~ new privileged ID information sid_h', both the privileged ID information sid_h and the new privileged ID information sid_h' corresponding to the tag ID information id_h, ~~the association of which with the privileged ID information sid_h is difficult to follow~~ in an updating section of the updater;

delivering the new privileged ID information sid_h' to the tag device from a second output section of the updater;

~~the tag device further~~

accepting an input of the new privileged ID information sid_h' at a second input section of the tag device; and

~~the read/write section of the tag device~~ storing the new privileged ID information sid_h' in the confidential value memory.

Claim 43 (Currently Amended): A tag privacy protection method for preventing privacy information of a user from being acquired from information which is delivered from a tag device, in which the privileged ID information sid_h which is a random value r_h related to each tag ID information id_h is stored in a confidential value memory of each tag device h ($h \in \{1, \ldots, m\}$, where m represents a total number of tag devices), and each tag ID information id_h and privileged ID information sid_h which is the random value r_h related to the tag ID information id_h are stored in a privileged ID memory of an updater which is provided externally of each tag device h in a manner relating to each other; comprising: ~~the steps of~~

~~the tag device h~~

reading out the privileged ID information sid_h stored in the confidential value memory thereof in a first read/write section of the tag device h;

~~and~~ delivering the privileged ID information sid_h to the updater from a first output section of the tag device h;

~~the updater~~

accepting an input of the privileged ID information sid_h at a first input section of the updater;

generating a new random value r_h' in a random value generator of the updater;

selecting tag ID information id_h corresponding to the privileged ID information sid_h which is accepted as the input from the privileged ID memory and storing the new random value r_h' in the privileged ID memory in a manner relating to new privileged ID information sid_h' in a second read/write section of the updater;

~~and~~ delivering the new privileged ID information sid_h' to the tag device h from a second output section of the updater;

~~the tag device h further~~

accepting an input of the new privileged ID information sid_h' at a second input section of the tag device h; and

~~the read/write section of the tag device~~ storing the new privileged ID information sid_h' in the confidential value memory.

Claim 44 (Currently Amended): A tag privacy protection method for preventing privacy information of a user from being acquired from information which is delivered from a tag device, in which privileged ID information sid_h is stored in a confidential value memory of each tag device h ($h \in \{1, \ldots, m\}$, where m represents a total number of tag devices), the privileged ID information sid_h including a first encrypted text according to a common key encryption technique which corresponds to each tag ID information id_h and key ID information kid_j of a common key k_j used in the encryption ($j \in \{1, \ldots, n\}$, where n represents a total number of tag devices), and each key ID information kid_j are stored and each common key k_j in a key memory of an updater which is provided externally of each tag device h in a manner relating to each other; comprising: ~~the steps of~~

~~the tag device h~~

reading out the privileged ID information sid_h stored in the confidential value memory thereof in a first read/write section of the tag device h;

~~and~~ delivering the privileged ID information sid_h to an updater from a first output section of the tag device h;

~~the updater~~

acceptings an input of the privileged ID information sid_h at a first input section of the updater;

extracting the common key k_j corresponding to the key ID information kid_j included in the privileged ID information sid_h from the key memory by a second read/write section of the updater;

decrypting the first encrypted text using the common key k_j extracted by the second read/write section to extract tag ID information id_h by an ID extractor of the updater;

generating a second encrypted text, ~~the association of which with the first encrypted text is difficult to follow,~~ using the tag ID information id_h extracted by the ID extractor and the common key k_j which is used in the extraction in an encryptor of the updater;

~~and~~ delivering new privileged ID information sid_h' including the second encrypted text and the key ID information kid_j of the common key k_j to the tag device h from a second output section of the updater;

~~the tag device h further~~

accepting an input of the new privileged ID information sid_h' at a second input section of the tag device h; and

~~the first read/write section of the tag device~~ storing the new privileged ID information sid_h' in the confidential value memory.

Claim 45 (Currently Amended): A tag privacy protection method for preventing privacy information of a user from being acquired from information which is delivered from a tag device, in which privileged ID information sid_h is stored in a confidential value memory of each tag device h ($h \in \{1, \ldots, m\}$, where m represents a total number of tag devices), the privileged ID information sid_h including a first encrypted text according to a pubic key encryption technique which corresponds to each tag ID information id_h and key ID

information kid_j for a key pair (sk_j, pk_j) (where sk_j represents a secret key and pk_j represents a public key, $j \in \{1, \ldots, n\}$, where n represents a total number of tag devices), and each key ID information kid_j and each key pair (sk_j, pk_j) in a key memory of an updater which is provided externally of each tag device h in a manner relating to each other; comprising: ~~the steps of~~

~~the tag device h~~

reading out the privileged ID information sid_h stored in the confidential value memory in a first read/write section of the tag device h;

~~and~~ delivering the privileged ID information sid_h to an updater from a first output section of the tag device h;

~~the updater~~

accepting an input of the privileged ID information sid_h at a first input section of the updater;

extracting the key pair (sk_j, pk_j) which corresponds to the key ID information kid_j which is included in the privileged ID information sid_h accepted as the input to the first input section by a second read/write section of the updater;

decrypting the first encrypted text using the secret key sk_j extracted by the second read/write section to extract the tag ID information id_h by an ID extractor of the updater;

generating a second encrypted text, ~~the association of which with the first encrypted text is difficult to follow,~~ using the tag ID information id_h extracted by the ID extractor and the public pk_j which is extracted by the second read/write section by an encryptor of the updater;

~~and~~ delivering new privileged ID information sid_h' including the second encrypted text and the key ID information kid_j of the key pair (sk_j, pk_j) to the tag device h from a second output section of the updater;

~~the tag device h further~~

accepting an input of the new privileged ID information sid_h' at a second input section of the tag device h; and

~~the read/write section~~ storing the new privileged ID information sid_h' in the confidential value memory.

Claim 46 (Currently Amended): A tag privacy protection method for preventing privacy information of a user from being acquired from information which is delivered from a tag device, in which privileged ID information sid_h is stored in a confidential value memory of each tag device h ($h \in \{1, \ldots, m\}$, where m represents a total number of tag devices), the privileged ID information sid_h including a first encrypted text according to re-encryptable public key encryption technique which corresponds to each tag ID information id_h and key ID information kid_j of the public key pk_j ($j \in \{1, \ldots, n\}$, where n represents a total number of keys), each key ID information kid_j and each public key pk_j are stored in a key memory of an updater which is provided externally of each tag device h in a manner relating to each other; comprising: ~~the steps of~~

~~the tag device h~~

readings out the privileged ID information sid_h stored in the confidential value memory in a first read/write section of the tag device h;

~~and~~ deliverings the privileged ID information sid_h to an updater from a first output section of the tag device h;

~~the updater comprising~~

accepting an input of the privileged ID information sid_h at a first input section of the updater;

extracting the public key pk_j which corresponds to the key ID information kid_j included in the privileged ID information sid_h which is accepted as the input to the first input section from the key memory by a second read/write section of the updater;

re-encrypting the first encrypted text in the privileged ID information sid_h using the public key pk_j extracted by the second read/write section to generate a second encrypted text, ~~the association of which with the first encrypted text is difficult to follow,~~ by an encryptor of the updater;

~~and for~~ delivering new privileged ID information sid_h' including the second encrypted text and the key ID information kid_j of the public key pk_j to the tag device h from a second output section of the updater;

~~the tag device h further~~

accepting an input of the new privileged ID information sid_h' at a second input section of the tag device h; and

~~the read/write section~~ storing the new privileged ID information sid_h' in the confidential value memory.

Claim 47 (Currently Amended): A tag privacy protection method for preventing privacy information of a user from being acquired from information which is delivered from a tag device, in which privileged ID information sid_h which has privileged each tag ID information id_h is stored in a confidential value memory of each tag device h ($h \in \{1, \ldots, m\}$, where m represents a total number of tag devices); comprising: ~~the steps of~~

~~the tag device h~~

reading out the privileged ID information sid_h stored in the confidential value memory by a first read/write section of the tag device h;

~~and~~ delivering the privileged ID information sid_h to a first updater which is provided externally of the tag device h from a first output section of the tag device h;

~~the first updater~~

accepting an input of the privileged ID information sid_h at a first input section of the first updater;

determining tag ID information id_h from the privileged ID information sid_h by an ID extractor of the first updater;

~~and~~ delivering the tag ID information id_h to a second updater which is provided externally of the tag device h from a second output section of the first updater;

~~the second updater~~

accepting an input of the tag ID information id_h at a third input section of the second updater;

generating new privileged ID information sid_h' which has privileged the tag ID information id_h by an encryptor of the second updater;

~~and~~ delivering the new privileged ID information sid_h' to the tag device h from a third output section of the second updater;

~~the tag device h further~~ accepting an input of the new privileged ID information sid_h' at a second input section of the tag device h; and

~~the read/write section~~ storing the new privileged ID information sid_h' in the confidential value memory.

Claim 48 (Original): An updater for updating privileged ID information in a tag device, the updater being provided externally of the tag device and comprising

a privileged ID memory for storing each tag ID information id_h and privileged ID information sid_h which is a random value r_h which corresponds to the tag ID information id_h in a manner relating to each other;

a first input section which accepts an input of the privileged ID information sid_h which is delivered from the tag device;

a random value generator for generating a new random value r_h';

a second read/write section connected to the privileged ID memory for selecting tag ID information id_h which corresponds to the privileged ID information sid_h which is accepted by the first input section as the input from the privileged ID memory and for relating this with the new random value r_h' as new privileged ID information sid_h' to be stored in the privileged ID memory;

and a second output section for delivering the new privileged ID information sid_h' to the tag device h.

Claim 49 (Currently Amended): An updater for updating privileged ID information in a tag device, the updater being provided externally of the tag device and comprising

a key memory for storing each key ID information kid_j ($j \in \{1, \ldots, n\}$, where n represents a total number of keys) and each common key k_j of a common key encryption technique in a manner relating to each other;

a first input section for accepting an input of privileged ID information sid_h which includes a first encrypted text according to the common key encryption technique which corresponds to the tag ID information id_h and key ID information kid_j of the common key k_j which is used in the encryption;

a second read/write section connected to the key memory for extracting the common key k_j which corresponds to the key ID information kid_j which is included in the privileged ID information sid_h from the key memory;

an ID extractor for decrypting the first encrypted text using the common key k_j which is extracted by the second read/write section to extract tag ID information id_h;

an encryptor for generating a second encrypted text, ~~the association of which with first encrypted text is difficult to follow,~~ using the tag ID information id_h extracted by the ID extractor and the common key k_j which is used in the extraction;

and a second output section for delivering new privileged ID information sid_h' which includes the second encrypted text and the key ID information kid_j for the common key k_j to the tag device h.

Claim 50 (Currently Amended): An updater for updating privileged ID information in a tag device, the updater being provided externally of the tag device and comprising

a key memory for storing each key ID information kid_j ($j \in \{1, \ldots, n\}$, where n represents a total number of keys) and each key pair (sk_j, pk_j) (sk_j represents a secret key and pk_j a public key) in a manner relating to each other;

a first input section for accepting an input of privileged ID information sid_h which includes a first encrypted text according to a public key encryption technique which corresponds to tag ID information id_h and key ID information kid_j for the public key pk_j which is used in the encryption;

a second read/write section connected to the key memory for extracting the key pair (sk_j, pk_j) which corresponds to the key ID information kid_j which is included in the privileged ID information sid_h accepted by the first input section as the input from the key memory;

an ID extractor for decrypting the first encrypted text using the secret key sk_j extracted by the second read/write section to extract tag ID information id_h;

an encryptor for generating a second encrypted text, ~~the association of which with the first encrypted text is difficult to follow,~~ using the tag ID information id_h extracted by the ID extractor and the public key pk_j extracted by the second read/write section;

and a second output section for delivering new privileged ID information sid_h' which includes the second encrypted text and the key ID information kid_j for the key pair (sk_j, pk_j) to the tag device h.

Claim 51 (Currently Amended): An updater for updating privileged ID information in a tag device, the updater being provided externally of the tag device and comprising

a key memory for storing each key ID information kid_j ($j \in \{1, \ldots, n\}$, where n represents a total number of keys) and each public key pk_j in a manner relating to each other;

a first input section for accepting an input of privileged ID information sid_h which includes a first encrypted text according to re-encryptable public key encryption technique which corresponds to tag ID information id_h and key ID information kid_j for the public key pk_j;

a second read/write section connected to the key memory for extracting the public key pk_j which corresponds to the key ID information kid_j which is included in the privileged ID information sid_h which is accepted by the first input section as the input from the key memory;

an encryptor for re-encrypting the first encrypted text which is included in the privileged ID information sid_h using the public key pk_j extracted by second read/write section to generate a second encrypted text~~, the association of which with the first encrypted text is difficult to follow~~;

and a second output section for delivering new privileged ID information sid_h' which includes the second encrypted text and the key ID information kid_j for the public key pk_j to the tag device h.

Claim 52 (Original): An updater according to one of Claims 49 to 51 in which the key ID information kid_j is information which is shared by a plurality unrelated tag devices.

Claim 53 (Currently Amended): An update solicitor for soliciting an updater to update privileged ID information in a tag device, the update solicitor being provided externally of the tag device and comprising

a privileged ID input section to which a plurality of kinds of privileged ID's, which are two or more kinds of re-encryptable encrypted texts corresponding to an identical tag ID information id_h, are input;

a privileged ID memory for storing a plurality of kinds of privileged ID's which are input thereto;

a privileged ID extractor connected to the privileged ID memory for extracting one of privileged ID's from the privileged ID memory at a given opportunity;

and a privileged ID output section for delivering the extracted privileged ID to the tag device.

Claim 54 (Currently Amended): A tag device for use in an automatic tag identification system comprising

a privileged ID input section to which a plurality of kinds of privileged ID's, which are two or more kinds of re-encryptable encrypted texts corresponding to an identical tag ID information id_h, are input;

a privileged ID memory for storing the plurality of kinds of privileged ID's which are input thereto;

a privileged ID extractor connected to the privileged ID memory for extracting one of the privileged ID's from the privileged ID memory at a given opportunity;

and a privileged ID output section for delivering the extracted privileged ID.

Claim 55 (Currently Amended): A tag privacy protection method for preventing privacy information of a user from being acquired from information which is delivered from a tag device, ~~in which a key ID and a key are stored in a key memory in a manner relating to each other, the tag device comprises a privileged ID memory including a read-only region in which a key ID is stored and a rewritable region in which a first privileged ID is stored~~; comprising: ~~the steps of~~

~~the tag device~~

extracting a~~the~~ key ID from a read-only region of a privileged ID memory of a tag device and extracting a~~and the~~ first privileged ID from a rewritable region of the privileged ID memory of the tag device by a read/write section of the tag device, the first privileged ID being an encrypted text of tag ID information, the encrypted text encrypted with a key specified by the key ID;

~~and~~ delivering the extracted key ID and the first privileged ID to an updater from a first output section of the tag device, the updater apparatus having a key memory storing a key ID and a key in a manner relating to each other;

~~the updater~~

accepting the key ID and the first privileged ID as inputs at a first input section of the updater;

extracting a key which corresponds to the key ID which is input to the first input section from the key memory by a first key extractor of the updater;

updating the first privileged ID to~~ng~~ a second privileged ID, the second privileged ID being a new encrypted text of the tag ID information corresponding to the first privileged ID, the new encrypted text encrypted with ~~the association of which with the first privileged ID is difficult to follow, using~~ the key extracted by the first key extractor, ~~and the first privileged ID which is input to the first input section~~ in a privileged ID updating section of the updater;

~~and~~ delivering the second privileged ID from a second output section;

~~the tag device further~~

accepting an input of the second privileged ID at a second input section of the tag device; and

~~the read/write section~~ storing the second privileged ID in the rewritable region of the privileged ID memory.

Claim 56 (Currently Amended): A tag privacy protection method according to Claim 55, further comprising: ~~the steps of~~

~~the updater additionally including a verification information generator which~~ generateings a verification information for the second privileged ID in a verification information generator of the updater apparatus;

~~the second output section of the updater~~ delivering the second privileged ID and the verification information from the second output section of the updater;

~~the second input section of the tag device~~ accepting the second privileged ID and the verification information as inputs at the second input section of the tag device;

~~the read/write section of the tag device~~ storing the second privileged ID and the verification information in the rewritable region the of privileged ID memory.

Claim 57 (Currently Amended): A tag privacy protection method according to Claim 55, further comprising: ~~the steps of~~

~~the read/write section of the tag device~~ extracting the key ID from the read-only region of the privileged ID memory and extracting ~~the~~a first or second ~~third~~ privileged ID from the rewritable region by the read/write section of the tag device;~~,~~

~~the first output section of the tag device~~ delivering the extracted key ID and the first or second ~~third~~ privileged ID to a decryptor from the first output section of the tag device;

~~the decryptor~~

accepting the key ID and the first or second privileged ID as inputs at a third input section of the decryptor;

extracting a key which corresponds to the key ID which is accepted by the third input section as an input from the key memory by a second key extractor of the decryptor;

decrypting the first or second privileged ID with ~~calculating an ID using the privileged ID which is input to the third input section and~~ the key extracted by the second key extractor of the decryptor to calculate the tag ID information in an ID calculator;

and verifying the structure of the calculated tag ID information by an ID structure verifier of the decryptor.

Claim 58 (Currently Amended): A tag device for use in an automatic tag identification system comprising

a privileged ID memory including a read-only region in which a key ID is stored and a rewritable region in which a first privileged ID is stored, the first privileged ID being an encrypted text of tag ID information, the encrypted text encrypted with a key specified by the key ID;

a read/write section for extracting the key ID and the first privileged ID from the privileged ID memory;

a first output section for delivering the key ID and the first privileged ID which are extracted; and

~~and~~ a second input section for accepting an input of a second privileged ID, the second privileged ID being a new encrypted text of the tag ID information corresponding to the first privileged ID, the new encrypted text encrypted with the key specified by the key ID, the first privileged ID differing from the second privileged ID ~~the association of which with the first privileged ID is difficult to follow~~;

wherein the read/write section storing the second privileged ID which is input in the rewritable region of the privileged ID memory.

Claim 59 (Original): A tag device according to Claim 58 in which the second input section additionally accepts an input of verification information for the second privileged ID and the read/write section additionally stores the verification information which is input in the rewritable region of the privileged ID memory.

Claim 60 (Currently Amended): A tag device according to Claim 58 in which each of the first and second privileged ID is encrypted text of ~~represent information which is~~ part of ~~information constituting~~ the tag ID information, the part of the tag ID information being ~~an ID and which is~~ inherent to each tag device~~, which is privileged alone~~.

Claim 61 (Original): A tag device according to Claim 58 in which an identical key ID is allocated to unrelated tag devices.

Claims 62-65 (Cancelled).

Claim 66 (Original): A computer readable record medium storing a tag program which enables a computer to function as a tag device according to one of Claims 17, 54 and 58.

Claim 67 (Original): A computer readable record medium storing a tag program which enables a computer to function as a backend apparatus according to Claim 18.

Claim 68 (Original): A computer readable record medium storing an update program which enables a computer to function as an updater according to one of Claims 48 to 51.

Claim 69 (Original): A computer readable record medium storing an update soliciting program which enables a computer to function as an update solicitor according to Claim 53.